Available online at www.sciencedirect.com

ScienceDirect

IERI Procedia 6 (2014) 22 – 32

Procedia IERI

www.elsevier.com/locate/procedia

2013 International Conference on Future Software Engineering and Multimedia Engineering

Hypergraph Reconfigurability Analysis

Ibtissam Abdesselam[a], Hafid Haffaf[a,]*

[a] Department of Computer Science, Oran University Es-Senia I.G.M.O, BP 15024 el menaouer Oran,, Algeria

Abstract

In the present paper, a hypergraph model for the structural system modeling and reconfigurability analysis has been presented. At first, we represent each system equation by a hyperedge, and then we extend the modeling hypergraph with others colored hyperedges (red and blue) which allows us to perform the analysis task. Based on the bottom up analysis hypergraph model, it's very easy to check the system reconfigurability in the presence of fault by verifying the existence of paths from the affected hyperedge to specifics blue hyperedges passing through specifics red hyperedges. The method is illustrated through a pedagogical example.

Peer review under responsibility of Scientific Committee of of Information Engineering Research Institute

Keywords: Reconfigurability analysis; Formal modeling; Hypergraphs; Faults Tolerant control.

1. Introduction

Over recent decades, in view of the enormous industrial systems complexity which can result in a large number of components that they contain, the problem of vulnerability to failures is posed. Therefore in order to ensure the main objectives of a system which are especially the production objectives, quality objectives and safety objectives, the development of fault tolerant system is therefore indispensable [1]. The latter requires two fundamental functions: a diagnostic operation (Fault detection and isolation FDI) processed in

* Corresponding author.
E-mail address: ibtissam-abdesselam@hotmail.fr (I.abdesselam), Haffaf.hafid@univ-oran.dz (H.Haffaf).

Peer review under responsibility of Scientific Committee of of Information Engineering Research Institute
doi:10.1016/j.ieri.2014.03.005

[15], [23] and the FTC (Fault tolerant control) operation which represents a necessary function to ensure the system functioning, and it consists either in the defects accommodation or reconfiguration (the case treated in this work). The development of FTC methods has been developed in many works such as: [22], [7].

The system reconfigurability analysis [13], [21] constitutes new trends towards fault-tolerance control systems. This can be classified into two different approaches: structural and functional. In this paper, we move towards the structural approach, where we propose a formal model which forms part of the graphical methods. These latter methods are generally offline used to check reconfigurability property in the sense of observability and controllability preservation yielding to detect at early stage whether the system could be go over the faults. Using developed and dedicated softwares in FDI-FTC domain which are based on graphical methodologies motivated us to generalize to hypergraph theory. The importance of the analysis hypergraph model lies on its structure, which allows us to represent with a hyperedge, the multi-dimensional constraints attached to each variables component, and that doesn't require a huge treatment in the reconfigurability analysis. Thus, considering that we haven't several faulty components at the same time, we define reconfigurability analysis through existence of paths in the defined analysis hypergraph.

The paper is organized in the following manner: the second section defines some basic hypergraph concepts and presents the different hypergraph systems modeling approaches. The section 3 indicates the hypergraph structural system modeling and its extension (the analysis hypergraph). This specific hypergraph is able to verify the structural fault tolerant property by using colored hyperedges. The methodology is illustrated by a pedagogical hydraulic system in the fourth section. In the last section, a short conclusion which summarizes the proposed approach is presented.

2. Hypergraph systems modeling

Any system S is defined as a combination of several components $\bigcup_{j=1}^{m} C_j$, the behavior (or the proper functioning) of each component $C_j \in S$ is described by a set of relationships F which are applied to a subset of variables of $X = \{x_1, x_2, ..., x_n\}$.

In the present paper, a physical system is represented through the mathematical object hypergraph. This latter can be considered as an appropriate tool to model the relationships associated to physical system constraints better than simple graph representation, and this is due to the hyperedge that can link more than two nodes.

2.1. Basic concepts of hypergraph

The present part shows some basic notions related to simple hypergraph, directed hypergraph and hyperpath.

Definition 1 In [6], Claude Berge has defined a hypergraph H (see Fig. 1 (a)) as any couple (V, E), such that:

- $V = \{v_i ; 1 \le i \le n\}$ is a finite set of vertices of H.
- $E = \{E_j ; 1 \le j \le m\}$ is a non empty set of hyperedges (edges in a graph) of H where $\bigcup_{j=1}^{m} E_j = V$.

Each hyperedge E_j is defined as a subset of vertices of H.

In a hypergraph H, two vertices v_{i_1}, v_{i_2} are called adjacent if they belong to the same hyperedge E_j.

Two hyperedges E_{j_1}, E_{j_2} are called adjacent if their intersection is not empty.

Definition 2 A directed hypergraph H or Dihypergraph (Fig. 1 (b) gives an example of an oriented hypergraph) is a hypergraph with oriented hyperedges (hyperarcs), This means that the extremities of a hyperedge have a very specific sense. A directed hyperedge is defined as a couple $E_j = (A, B)$, where A and B are two disjoint sets of vertices. The set A represents the hyperedge tail denoted by $T(E_j)$ and B his head $H(E_j)$.

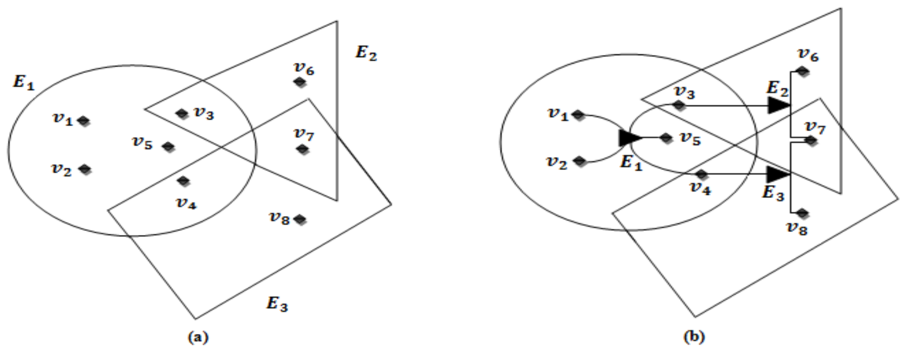

Fig. 1. (a) Hypergraph composed of three hyperedges E_1, E_2 and E_3 ; (b) Oriented hypergraph.

Definition 3 A hyperpath $HP_{s,t}$ from the source s to the destination t is a succession of vertices and hyperedges $HP_{s,t} = (v_{i_1} = s, E_{j_1}, v_{i_2}, E_{j_2}, .., E_{j_h}, v_{i_{h+1}} = t)$ where:

$$s \in T(E_{j_1}), t \in H(E_{j_h}) \wedge v_{i_g} \in H(E_{j_{g-1}}) \cap T(E_{j_g}), 2 \leq g \leq h. \tag{1}$$

The previous general definitions regarding the directed hypergraph are given in [12] and [3].

2.2. System modeling with hypergraph

In the last decade, the hypergraph becomes effective analysis tool in different computer science domains, it's used in the peer to peer context for representing the peers and their relationships [16], in the clustering area [10] by modeling the clusters with vertices and clusters of clusters with hyperedges, and in image processing in the work [20], which is focused on the determination of the properties resulting from the hypergraphs theory and on the analysis of their adequacy with image problems, particularly edge and noise detection.

In addition to simple hypergraphs, directed hypergraphs have also been used to model other problems such as: the production and manufacturing system [11] by representing each production system activity with a hyperedge, linking the inputs (consumed goods) to the outputs (produced goods), the propositional logic [2] for the reason to model each propositional symbol with a node, and each Horn clause with a hyperedge, such that the left side of the clause is the tail of the hyperedge and the right side the head, and in RDF documents representation [18]. The directed hypergraph associated to a RDF graph T is defined as a triple $H(T)=(W,E,\rho)$, such that W is the set of nodes representing the resources of T (subject, object, or property), E the hyperedges associated to the triples (s, p, o) of T and ρ the role function of nodes that can take one of the value {'s', 'p', 'o'}.

3. Contribution to fault tolerance analysis

Up to present, in the literature in diagnosis and fault tolerance domains, for the purpose of minimizing as much as possible the major inconvenience of analytical methods, that concerns the requirement of the knowledge of system parameters which are not always available, several graphical based approaches have been proposed: [8] presents a sensor classification for the fault detection and isolation through a structural approach, in [5], [9], the authors have used the bonds graph in order to design a diagnosis method and in [19] a causal graph (digraph) is used to find residual set and diagnostic relation.

With the aim of establishing a fault tolerant control analysis within the meaning of system reconfiguration, and that intends to check at early stage the observability and controllability properties preservation in the presence of fault, we introduce in this section a hypergraph for structural system modeling, then we extend this latter with colored hyperedges that stand for their generation on the structural properties of the modeling hypergraph.

3.1. Modeling hypergraph

In this paper, the hypergraph is used for modeling the mathematical equations system, where the choice of this tool is justified by mainly two facts: (i) the graphical modeling does not require a detailed knowledge of the system parameters, and describes the system structure by the existence or not of the link between variables and constraints, (ii) the hyperedges related to the hypergraph consisting of several nodes (n-ary relation) can be used to represent the constraints associated to the system components.

The idea of hyperedge system constraint modeling was inspired from [17], which defined systems of systems (a set of independent and interconnected systems) as a constraints problem. The variables represent the low level systems and the constraints the relations between systems from different levels. Consequently, the authors have used the hyperedges to model the sub systems, and the vertices for the elementary components.

According to our approach, a system can be modeled by a hypergraph $H=(V,E)$, where:

- The vertices V corresponding to a finite set of variables $X=K+L$, where $K=U\cup Y$ represents a set of known variables (measured and control) and the set L, is the unknown variables (internal variables, unknown inputs and disturbances).
- The hyperedges E that model the mathematical equations system, such that each hyperedge $E_j \in E$ is defined for a constraint C_j.

3.2. Analysis hypergraph

In [4], the graphical observability, controllability and reconfigurability analysis requires a huge treatment in each analysis step to check the existence of paths from unknown to known variables.

Due to increasing complexity in large scale systems, the topology, the combinatorial structures, the mutli dimensional relationships requires hypergraph representation. Therefore in order to remedy all this kinds of problems, the modeling hypergraph extension has been proposed. The advantage of the latter is that require little treatment in the reconfigurability analysis, and it needs only one course of the modeling hypergraph for determining the accessibility of the unknown variables by the known ones, and this is in the generation of the colored hyperedges. The structure of the extended hypergraph (analysis hypergraph) is simple, where it has a smaller size even for the large systems. Indeed, several paths in simple graph could be compacted by hyperedge pictorial representation.

The analysis hypergraph proposed in our work was inspired from the AND/OR tree, where the AND nodes correspond to hyperedges with red color, OR nodes blue hyperedges, and the leaves the system components constraints.

Defintion 4 An analysis hypergraph $H = (E, V)$ is an extension of the modeling hypergraph where:

- $V = \{v_i; 1 \le i \le n\}$ is a finite set of variables.
- $E = \varepsilon_1 \cup \varepsilon_2 \cup \varepsilon_3$ where $\varepsilon_1 = \{E_j; 1 \le j \le m\}$ is the set of hyperedges modeling of system components constraints, $\varepsilon_2 = \{Er_k; 1 \le \mathrm{k} \le g\}$ is the set of red hyperedges or *And* hyperedges (represented with a dotted line), $\varepsilon_3 = \{Eb_l; 1 \le l \le p\}$ is the set of blue hyperedges (represented with a dashed line) and they are called also *Or* hyperedges.

Definition 5 An elementary hyperegde E_j is the modeling of a system component constraint.

Definition 6 An observed (controlled) hyperedge E_j is a hyperedge containing an output (input) variable.

The generation of the colored hyperedges of analysis hypergraph is performed straightforwardly from the directed modeling hypergraph. Therefore the modeling hypergraph orientation is a necessary step, where for each hyperedge E_j that models a constraint C_j, the definition of a left side variables of C_j as $T(E_j)$ and the right side $H(E_j)$, this orientation manner was inspired from [4].

Property 1. In an analysis hypergraph *H*, a red hyperedge Er_k is a hyperedge associated to a hyperpath $HP_{s,t}$ connecting a state variable x_i to a known variable *K* where:

$$Er_k = \bigcup_{h=j_1}^{j_g} E_h \,/\, E_h \in HP_{s,t}. \tag{2}$$

Example For illustration, consider the Fig. 1(b) example, where we assume that the vertice v_3 is an unknown variable and v_8 is an output variable. Then the red hyperedge of the hyperpath HP_{v_3,v_8} is shown in the Fig. 2:

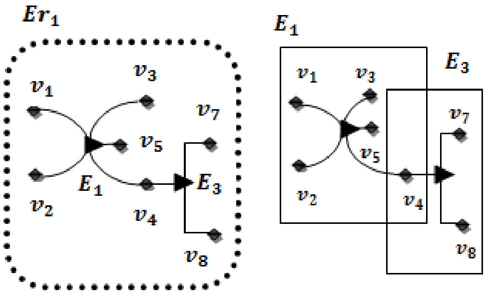

Fig. 2. Red hyperedge Er_1 composed of two elementary hyperedges E_1 and E_3 .

Property 2. The blue hyperedge Eb_l of the analysis hypergraph H is the combination of a subset of red hyperedges Er_k generated for the same state variable x_i such that:

$$Eb_1 = \bigcup_{h=k_1}^{k_g} Er_h \qquad (3)$$

Example Consider the same system of the Fig. 1(b), and assume that the vertice v_6 is an output variable. Then for the two hyperpaths HP_{v_3,v_8}, HP_{v_3,v_6}, two red hyperedges Er_1, Er_2 associated to the variable v_3 can be generated and grouped in a blue hyperegde Eb_1 as shown in the Fig. 3.

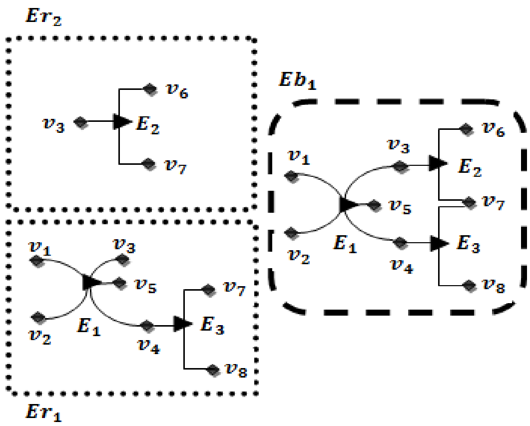

Fig. 3. Blue hyperedge Eb_1 composed of two red hyperedges Er_1 and Er_2.

3.3. Construction steps of the analysis hypergraph

In order to construct the analysis hypergraph, the following steps should be applied:

- Step1: Point to a state variable x_i.
- Step2: Verify the structural properties of x_i from the directed hypergraph modeling (observability and controllability).
- Step3: If the step2 is verified, that is to say that there is one or more hyperpaths $HP_{s,t}$ connecting the variable x_i to a known variable *K*, then pass to the Step4. Else increment the value of *i* and return to the step1.
- Step4: For each hyperpath $HP_{s,t}$, collect the set of elementary hyperedges $\varepsilon_{1_1} \subset \varepsilon_1$ contained in $HP_{s,t}$ in a red hyperedge Er_k.
- Step5: In a blue hyperedge Eb_l, we combine the red observed hyperedges $\varepsilon_{2_1} \subset \varepsilon_2$ obtained from the previous step (if their number is more than one hyperedge), and we add also the red controlled hyperedges $\varepsilon_{2_2} \subset \varepsilon_2$ if they exceed one hyperedge. The increment of the *i* value is necessary before to return to the first step.

3.4. Analysis of fault tolerance using the hypergraph analysis

The system reconfigurability analysis represents the first step of the development of fault tolerant control, i.e the ability of the system to continue achieving its tasks even in the presence of failure. According to the works [14] and [4], the reconfigurability or fault tolerance property can be defined as follows:

Definition 7 The system is called reconfigurable in the presence of fault, if the observability and controllability properties of the nominal system are preserved in the faulty one.

In this work, for robust fault tolerant system design, the verification of the reconfigurability property for instance are performed offline, i.e not during system functioning. So based on the hierarchical representation of the analysis hypergraph, we can deduce whether the considered system is fault tolerant in the presence of a faulty component, if its variables remain observable and controllable.

Definition 8 In the hierarchical analysis hypergraph representation, A path $P_{s,t}$ is a sequence of hyperedges $P_{s,t} = \left(E_{z_1} = s, E_{z_2}, ..., E_{z_h} = t\right)$, such that:

$$E_{z_k} \subset E_{z_{k+1}}, k = 1, ..., h-1 \tag{4}$$

It should be noted that in the present paper, the representation Ef_j design the elementary hyperedge affected by the fault, which is issued from the faulty component, and the symbol $\Rightarrow$ an association relation between state variable and colored hyperedge.

Considering that the system does not contain differential cycles, the following proposition is the graphic reconfigurability condition.

Proposition 1. The system reconfigurability property is preserved after the failure of a component if and only if:

$$\forall x_i \in Ef_j, \forall Er_k \in \varepsilon_2 / Er_k \Rightarrow x_i \wedge Ef_j \subseteq Er_k, \exists P_{s,t} / s = Er_k, t = Eb_l \in \varepsilon_3 \wedge Eb_l \Rightarrow x_i \tag{5}$$

Contrary to the approaches that require for checking the preservation of observability and controllability properties in the presence of fault, to determine the accessibility of each state variable by two different known variables (input and output), the proposed approach consists to verify only the observability, controllability properties of the variables attached to the affected hyperedge, ie: the integration of each red hyperedge Er_k begins by the affected hyperedge Ef_j in blue hyperedge Eb_l, such that Er_k and Eb_l are associated to the same state variable x_i.

4. Illustrative example

In order to show the impact of our approach, we have applied in this section the hypergraph modeling and analysis presented in the previous sections on an example. This example (see the Fig. 4) is a system with three coupled water tanks.

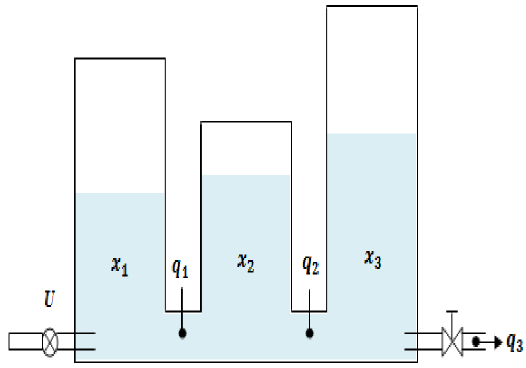

Fig. 4. Three coupled water tanks process.

This system is described by the following dynamics equations:

- $C_1 : \dot{x}_1 = u - q_1$
- $C_2 : \dot{x}_2 = q_1 - q_2$
- $C_3 : \dot{x}_3 = q_2 - q_3$
- $C_4 : q_1 = x_1 - x_2$
- $C_5 : q_2 = x_2 - x_3$
- $C_6 : y_1 = x_1$
- $C_7 : y_2 = q_3$

where x_i are tank levels, q_i flows, y_i measurements and u known control input. According to the previous constraints C_6 and C_7, it's clear that the available measurements are the level x_1 in the first tank 1 and the flow q_3.

The corresponding hypergraph system modeling is shown in Fig. 5.

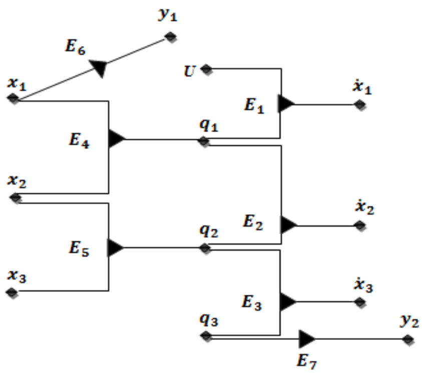

Fig. 5. Oriented hypergraph modeling of the three coupled water tanks system.

After having applied the first four construction steps previously mentioned to the variable x_2, we have arrived to generate the following result:

- The red observed hyperedge $Er_4 = E_4 \cup E_6 = \{x_2, x_1, q_1, y_1\}$ since there exists a hyperpath $HP_{s,t}$ from the variable x_2 to the known (observed) variable y_1.
- The red observed hyperedge $Er_9 = E_5 \cup E_3 \cup E_7 = \{x_2, x_3, q_2, q_3, \dot{x}_3, y_2\}$.
- The red controlled hyperedge $Er_6 = E_4 \cup E_1 = \{x_2, x_1, q_1, \dot{x}_1, u\}$.
- The red controlled hyperedge $Er_{10} = E_5 \cup E_2 \cup E_1 = \{x_2, x_3, q_2, q_1, \dot{x}_2, \dot{x}_1, u\}$.

Since the previous set of hyperedges generated for the state variable x_2 contains two red observed hyperedge Er_4 and Er_9, and two red controlled hyperedges Er_6 and Er_{10}, then the generation of the blue hyperedges Eb_2 is therefore necessary such that:

- $Eb_2 = Er_4 \cup Er_9 \cup Er_6 \cup Er_{10} = \{x_2, x_3, x_1, q_1, q_2, q_3, \dot{x}_1, \dot{x}_2, \dot{x}_3, y_1, y_2, u\}$.

The reconfigurability property analysis is the next phase after the termination of the analysis hypergraph construction. Through the bottom up analysis hypergraph, the reconfigurability property can be easily checked by hand (the existence of opportunities of reconfiguration is related to the existence of paths from the affected elementary hyperedges to specifics blue hyperedges), however the rank computation of the observability and controllability matrices (analytical approaches) is a very difficult manual task.

With the aim to clarify the graphic reconfigurability condition, let us suppose that a fault f_1 can affect the first sensor, therefore in the presence of this fault, the constraint C_6 describes the proper functioning of the sensor becomes incorrect. So for checking if the system remain observable and controllable after this fault, it is necessary to verify the observability and controllability properties of the variable x_1 (the only variable associated to the affected elementary hyperedge Ef_6), this is carried out by ascending hypergraph analysis as shown in Fig. 6.

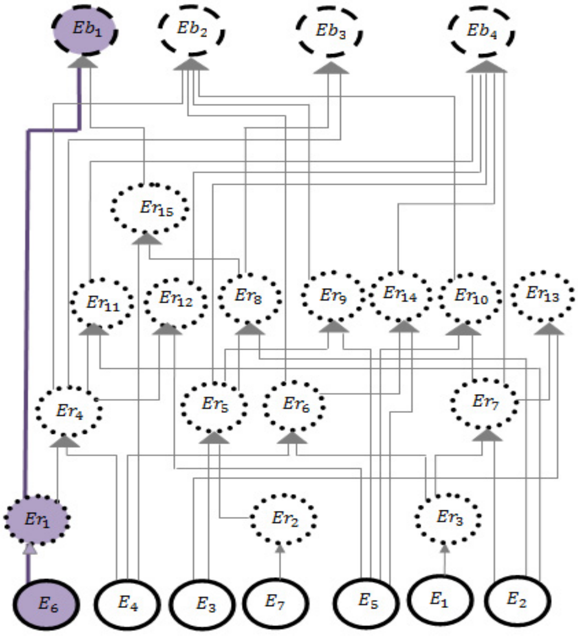

Fig. 6. Hierarchical representation of the analysis hypergraph.

Since the faulty component is a sensor, therefore it is sufficient to verify only the observability property of the observed variables (in this case the variable x_1). Performing an ascending verification of the analysis hypergraph, we are fined that there exists a path $P_{s,t}$ from the affected elementary hyperedge Ef_6 to a blue observed hyperedge Eb_1 generated for the state variable x_1, which implies that the system observability property is verified in the supposed fault.

5. Conclusion

For robust fault tolerant system design, the reconfigurability study is considered as the first development step. In this paper, we proposed a new hypergraph model defined by colored hyperedges, which is simultaneously used for modeling and reconfigurability analysis. This specific hypergraph is a powerful methodology that allows to better visualizing certain structural system properties.

Through the bottom up analysis hypergraph, it's very simple to check if a system described by a set of dynamic mathematical equations is reconfigurable in the presence of faulty component, and this is due by verifying the existence of paths from the affected elementary hyperedge to specifics blue hyperedges.

Among the perspectives that we plan, is the generalization of our approach for the case of presence of several faulty components, and the application to more real complex industrial application.

References

[1] Assas L.M. Analyse de la tolérance aux fautes: approche fonctionnelle et structurelle. PhD thesis, Université de Lille, 2002.
[2] Giaccio R, Ausiello G. On-line algorithms for satisability formulae with uncertainty. Theoretical Computer Science, 171: 3–24, 1997.
[3] Nanni U, Ausiello G, Italiano F.G. Hypergraph traversal revisited: Cost measures and dynamic algorithms. In Symposium on Mathematical Foundations of Computer Science, Springer-Verlag, 1450: 1–16, 1998.
[4] Belkherchi N. Contribution à l'étude du diagnostic et de la commande tolérante aux fautes par l'approche structurelle: Application aux procedes biologiques. PhD thesis, Universite de Toulouse 3 Paul Sabatier, 2011.
[5] Merzouki R, Benmoussa S, Ould-Bouamama B. Bond graph approach for plant fault detection and isolation: Application to intelligent autonomous vehicle. IEEE Transactions on Automation Science and Engineering, 2013.
[6] Berge C. Graphs and Hypergraphs, 1973.
[7] Wu N, Blanke M, Starosweicki M. Concepts and methods in fault-tolerant control. American Control Conference, Arlington, USA, 4: 2606–2620, 2001.
[8] Trinh D-Do.T, Commault C, Dion J. Sensor classification for the fault detection and isolation, a structural approach. International Journal of Adaptive Control and Signal Processing, 25: 1–17, 2011.
[9] Dauphin-Tanguy G, Djeziri M, Ould Bouamama B. Bond graph model based robust control and diagnosis. Book Chapter in Springer Verlag, ISBN: 1848828810, 662 pages, 2010.
[10] Suzuki E, Durand N, Crmilleux B. Visualizing transactional data with multiple clusterings for knowledge discovery. ISMIS, 2006.
[11] Grazia scutella M, Gallo G. Directed hyper-graphs as a modeling paradigm, 21: 97–123, 1998.
[12] Nguyen S, Pallottino S, Gallo G, Longo G. Directed hypergraphs and applications. Discrete applied mathematics, 42: 177–201, 1993.
[13] Starosweicki M, Gehin A. Reconfiguration analysis using generic component models. IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans 38, 3: 575–583, 2008.

[14] Staroswiecki M, Gehin A, Assas M. Structural analysis of system reconfigurability. In Proceedings of IFAC/Safeprocess, 1: 292–297, 2000.
[15] Isermann R. Model-based fault detection and diagnosis-status and applications. Annual Reviews in Control, 29: 71–85, 2005.
[16] Durand N, Hajjar M, Ismail A, Quafafou M. An efficient architecture for information retrieval in p2p context using hypergraph. International Journal of Computer Networks & Communications, 3: 166, 2011.
[17] Khalil W, Merzouki R, Ould-Bouamama B, Haffaf H. Hypergraph models for system of systems supervision design. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 42: 1005–1012, 2012.
[18] Martnez-Morales A, Vidal M. Directed hyper-graphs for Rdf documents, 33: 59 – 67, 2010.
[19] Ostasz A. Causal graph and its application to finding residual set and diagnostic relation. PhD thesis, Warsaw University of Technology, 2006.
[20] Soufiane R. Hypergraphe de Voisinage Spatiocolorimetrique. Application en traitement d'images: Detection de contours et du bruit. PhD thesis, Universite de Bourgogne Dijon, 2004.
[21] Starosweicki M. On reconfigurability with respect to actuator failures. IFAC Safeprocess Barcelona, 2002.
[22] Gehin A.L, Starosweicki M. From control to supervision. In Annual Reviews in Control, 25: 1–11, 2001.
[23] Yassine A. Génération des tests et placement de capteurs pour le diagnostic des systèmes physiques s'appuyant sur une modélisation structurelle. PhD thesis, Laboratoire G-SCOP de Grenoble, 2008.